

S1 – M1104 – Introduction aux bases de données

Devoir n° 3 – Éléments de correction

1^{re} partie : ACSI

1. Analyse et conception des systèmes d'information.
2. Une contrainte référentielle permet d'indiquer, dans un schéma relationnel, que toute valeur d'un attribut « clé étrangère » doit apparaître comme valeur de l'attribut « clé primaire » correspondant. Par exemple :

Musiciens (Code_Musicien, Nom_Musicien, Code_Pays)

Pays (Code_Pays, Nom_Pays)

Musicien.Code_Pays référence Pays.Code_Pays

Toute valeur – par exemple 24 pour le tuple (137, 'Debussy', 24) – apparaissant dans la colonne Code_Pays de la table Musiciens doit exister comme clé primaire dans la table Pays – par exemple dans le tuple (24, 'France').

3. DEPARTEMENTS (numD, nomD)
ETUDIANTS (numD, numE, nomE)
ETUDIANTS.numD référence DEPARTEMENTS.numD
DEPARTEMENTS.numD \subseteq ETUDIANTS.numD
MATIERES (numM, nomM)
PLAGES (numP, débutP, finP)
SUIVRECOURS (numD, numE, numM, numP)
SUIVRECOURS.(numD, numE) référence ETUDIANTS.(numD, numE)
ETUDIANTS.(numD, numE) \subseteq SUIVRECOURS.(numD, numE)
SUIVRECOURS.numM référence MATIERES.numM
SUIVRECOURS.numP référence PLAGES.numP

2^e partie : Bases de données – Conception de requêtes SQL

Requête 1.

```
SELECT Film.Titre_Film, COUNT(DISTINCT Interprétation.code_personne) as nbAuteur
FROM Film
LEFT OUTER JOIN Interprétation ON Film.Code_film = Interprétation.Code_film
WHERE Film.Date_Sortie = 2010
GROUP BY Film.Code_Film, Film.Titre_Film
```

Remarques. Le GROUP BY Code_Film permet de conserver « sans les mélanger » les films ayant même titre. Il n'est pas nécessaire ici d'utiliser la table Personnes car la table Interprétation contient Code_Personne.

Requête 2.

```
SELECT Nom_Personne, Prénom_Personne, Titre_Oeuvre
FROM PERSONNE
INNER JOIN Réaliser ON Réaliser.Code_Personne = Personne.Code_Personne
INNER JOIN Film ON Réaliser.Code_Film = Film.Code_Film
INNER JOIN Adapter ON Adapter.Code_Film = Film.Code_Film
INNER JOIN Oeuvre on Oeuvre.Code_Oeuvre = Adapter.Code_Oeuvre
INNER JOIN Écrire ON Écrire.Code_Oeuvre= Oeuvre.Code_Oeuvre
AND Écrire.Code_Personne = Personne.Code_Personne
```

Remarque. Il n'est pas nécessaire ici d'utiliser la table Personne une deuxième fois, car la table Écrire contient le Code_Personne (par contre, si on l'utilise deux fois, il est nécessaire de la renommer via un « as auteur » par exemple).

Requête 3. Il est nécessaire de créer deux tuples, un dans la table Oeuvre, puis un dans la table Écrire :

```
INSERT INTO Oeuvre (Titre_Oeuvre, Date)
VALUES ('Les animaux fantastiques', 2001);
INSERT INTO ECRIRE (Code_Personne, Code_Oeuvre)
VALUES( (SELECT Code_Personne FROM Personne WHERE Nom_Personne = 'Rowling'),
        (SELECT Code_oeuvre FROM Oeuvre WHERE Titre_Oeuvre = 'Les animaux fantastiques') )
```

Requête 4.

```
UPDATE Réaliser  
    SET Code_Personne = (SELECT Code_Personne FROM Personne  
                        WHERE Nom_Personne = 'Raguéba' AND Prénom_Personne='Caroline')  
WHERE Code_Film = (SELECT Code_Film FROM Film  
                  WHERE Titre_Film = 'Le carnet de la jungle')
```

Requête 5.

```
DELETE FROM Pays
WHERE Code_Pays NOT IN (SELECT Code_Pays FROM Villes )
      AND Code_Pays NOT IN (SELECT Code_Pays FROM Personne )
```

ou bien :

```
DELETE FROM Pays  
WHERE Code_Pays NOT IN (SELECT Code_Pays FROM Villes  
                        UNION SELECT Code_Pays FROM Personne )
```